

Elasticsearch & Docker

Running High Performance
Fault Tolerant
Elasticsearch Clusters On Docker

Rafał Kuć – Sematext Group, Inc.

[@kucrafal](#) [@sematext](#) [sematext.com](#)

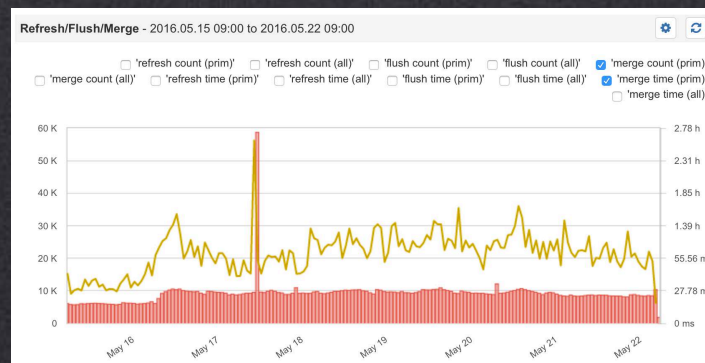
sematext



Next 20 minutes



elasticsearch

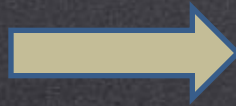


metrics

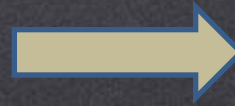
You Are Probably Familiar With That

```
itemprop="headline"><a href="<?php the_per
time itemprop="datePublished" datetime="<?
(categories = get_the_category()); the
attachment_id = get_field('image_slide
$separator = " // dimensions
$output = wp_get_attachment_image_src($
if (@categories) {
    $url = get_field('url');
    $title = get_field('title');
    $description = get_field('description');
    echo <div class="item">
        <h3><h3>
        <p>
    }
}
</div>
</pre>
```

Development

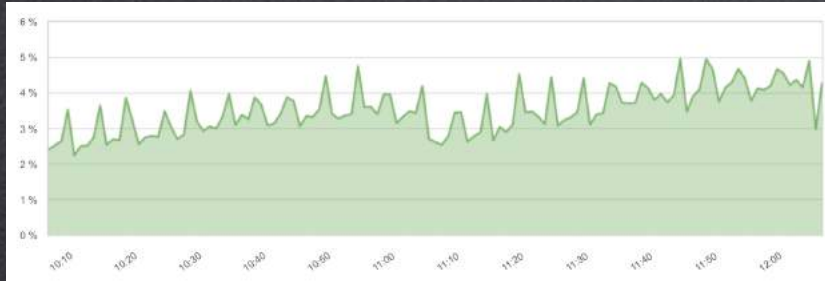


Test



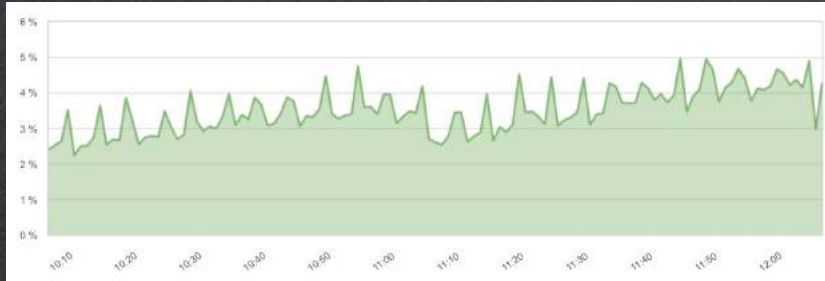
QA

And Problems That Come With It

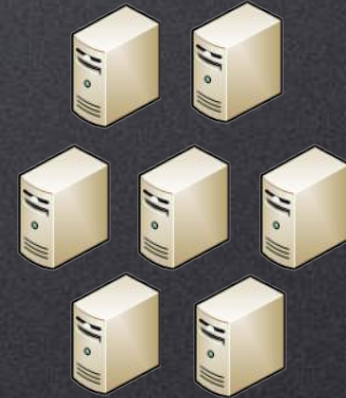


Resources not utilized

And Problems That Come With It

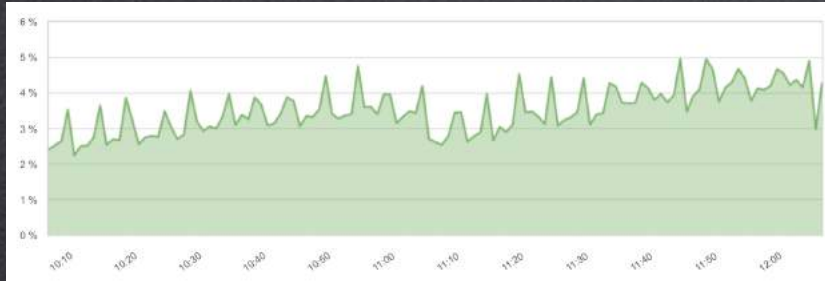


Resources not utilized

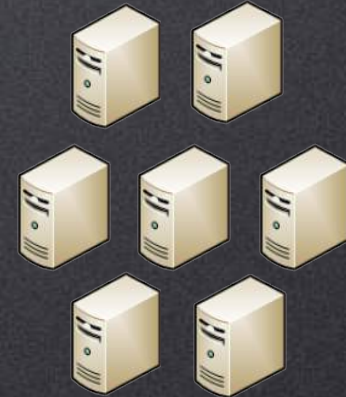


Overprovisioned Servers

And Problems That Come With It



Resources not utilized

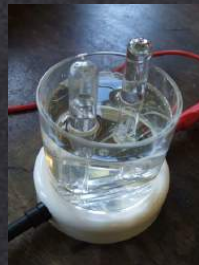


Overprovisioned Servers

```
itemprop="id" data-bbox="104 687 231 813">
<div data-bbox="104 687 231 813" data-label="Code-Block">
</div>

```

≠



≠



Why Docker?



Light weight

Based on
Open Standards



Secure

Containers vs Virtual Machines



Hardware

Traditional Virtual Machine

Containers vs Virtual Machines

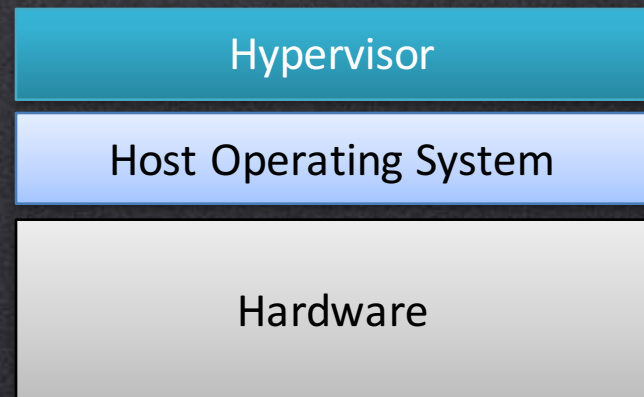


Host Operating System

Hardware

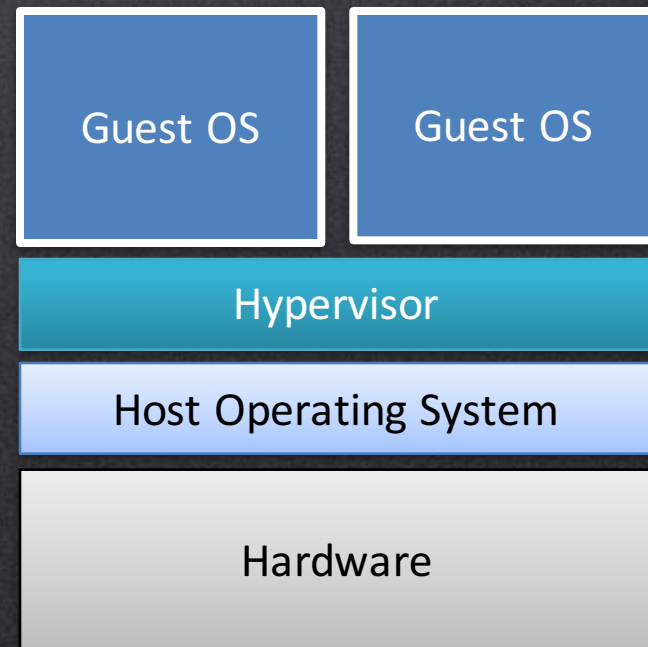
Traditional Virtual Machine

Containers vs Virtual Machines



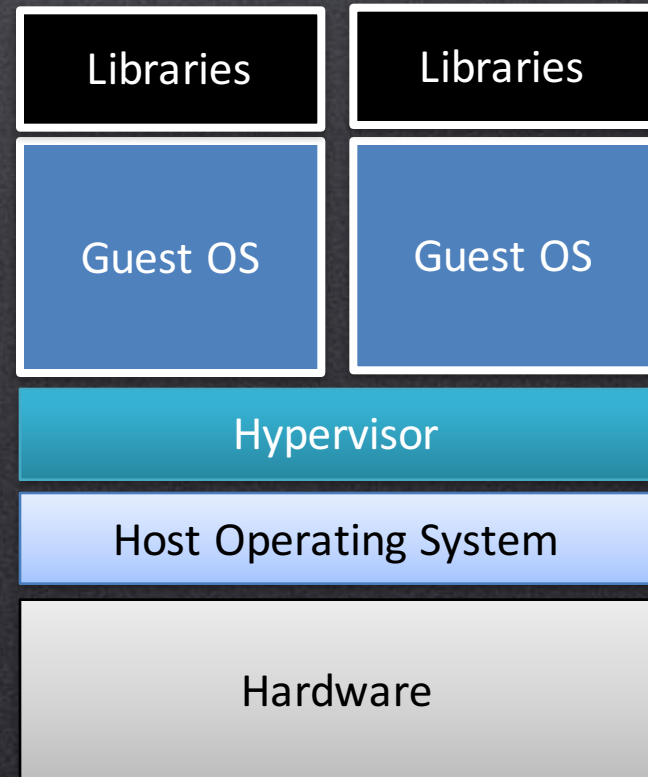
Traditional Virtual Machine

Containers vs Virtual Machines



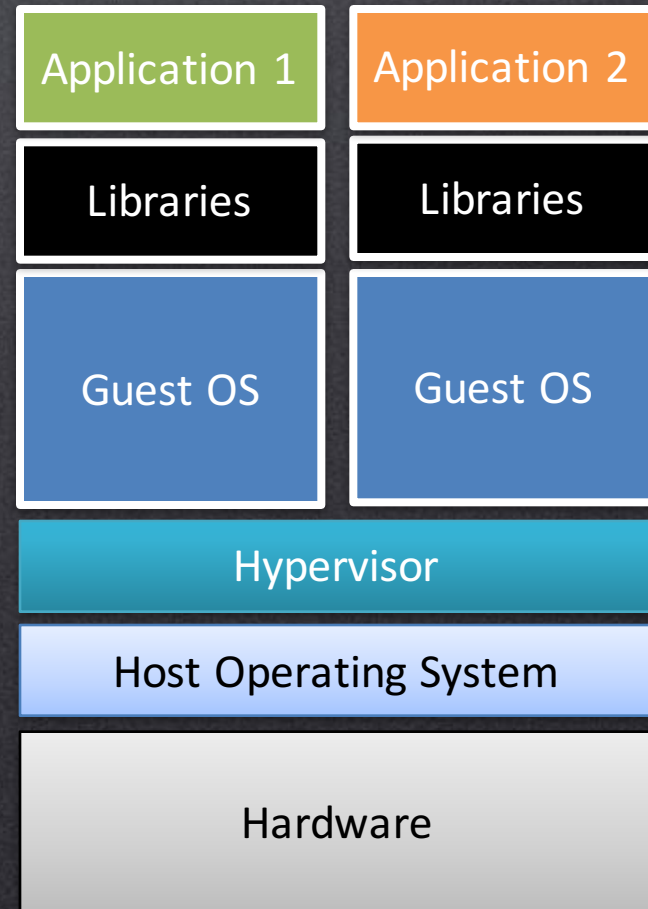
Traditional Virtual Machine

Containers vs Virtual Machines



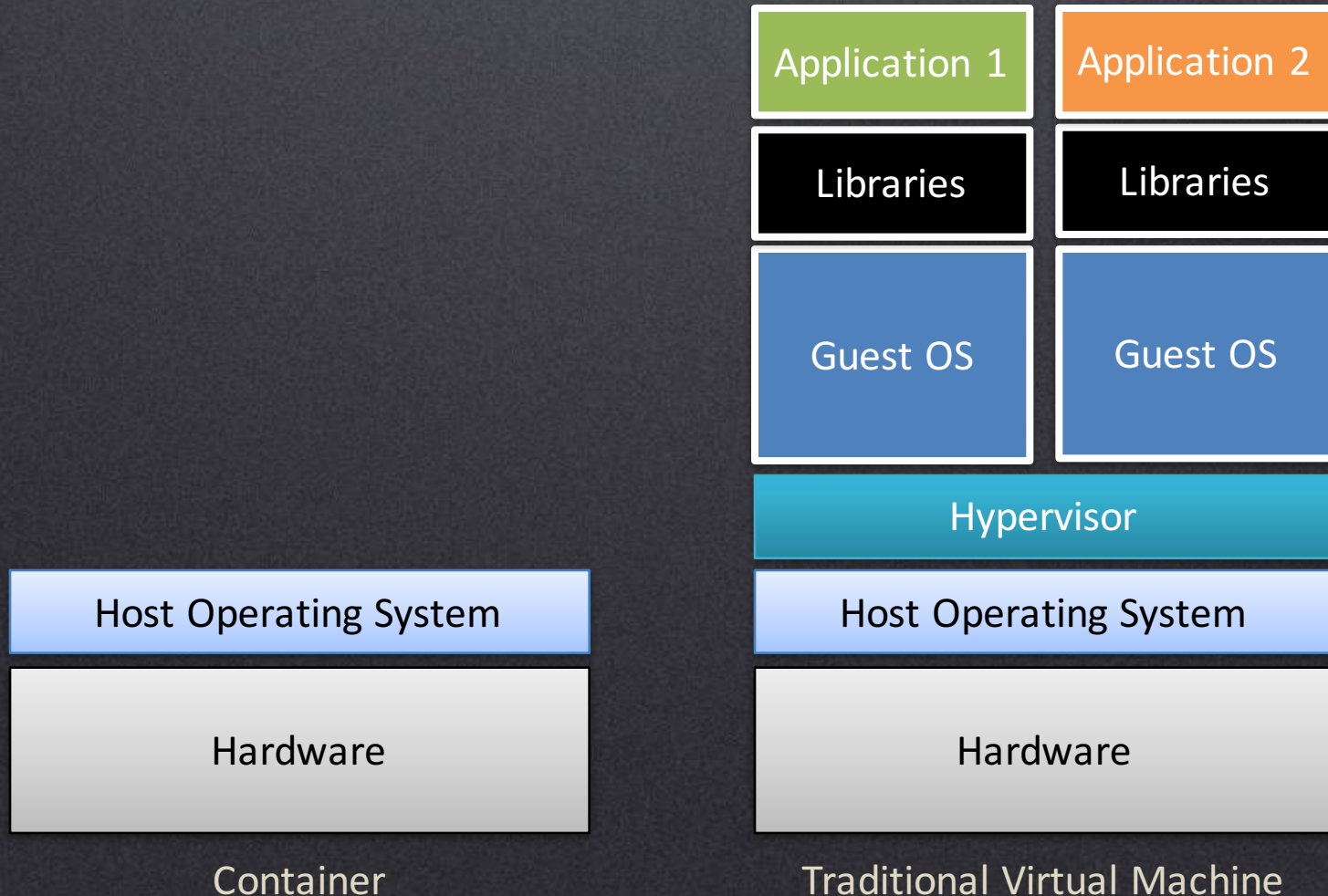
Traditional Virtual Machine

Containers vs Virtual Machines

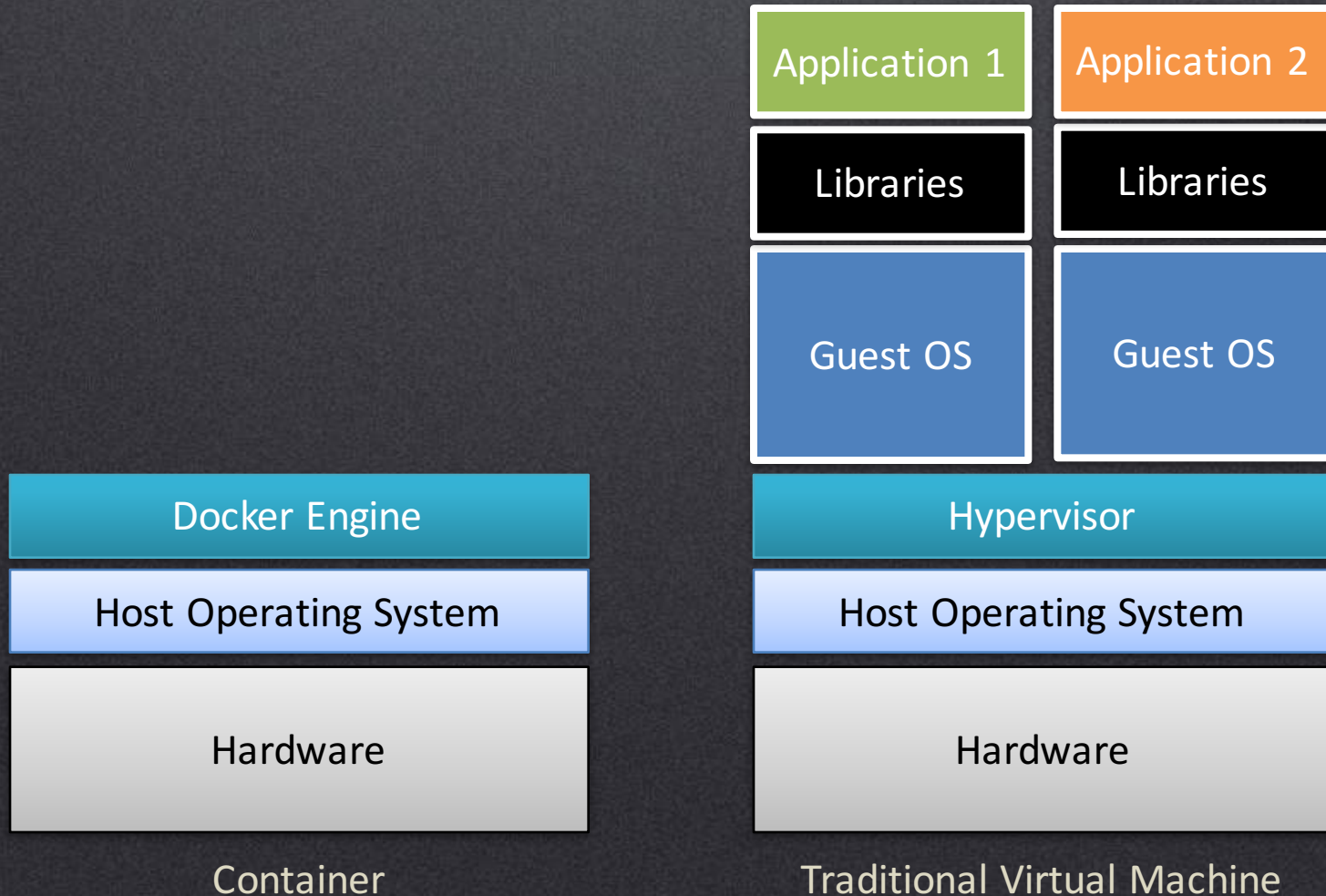


Traditional Virtual Machine

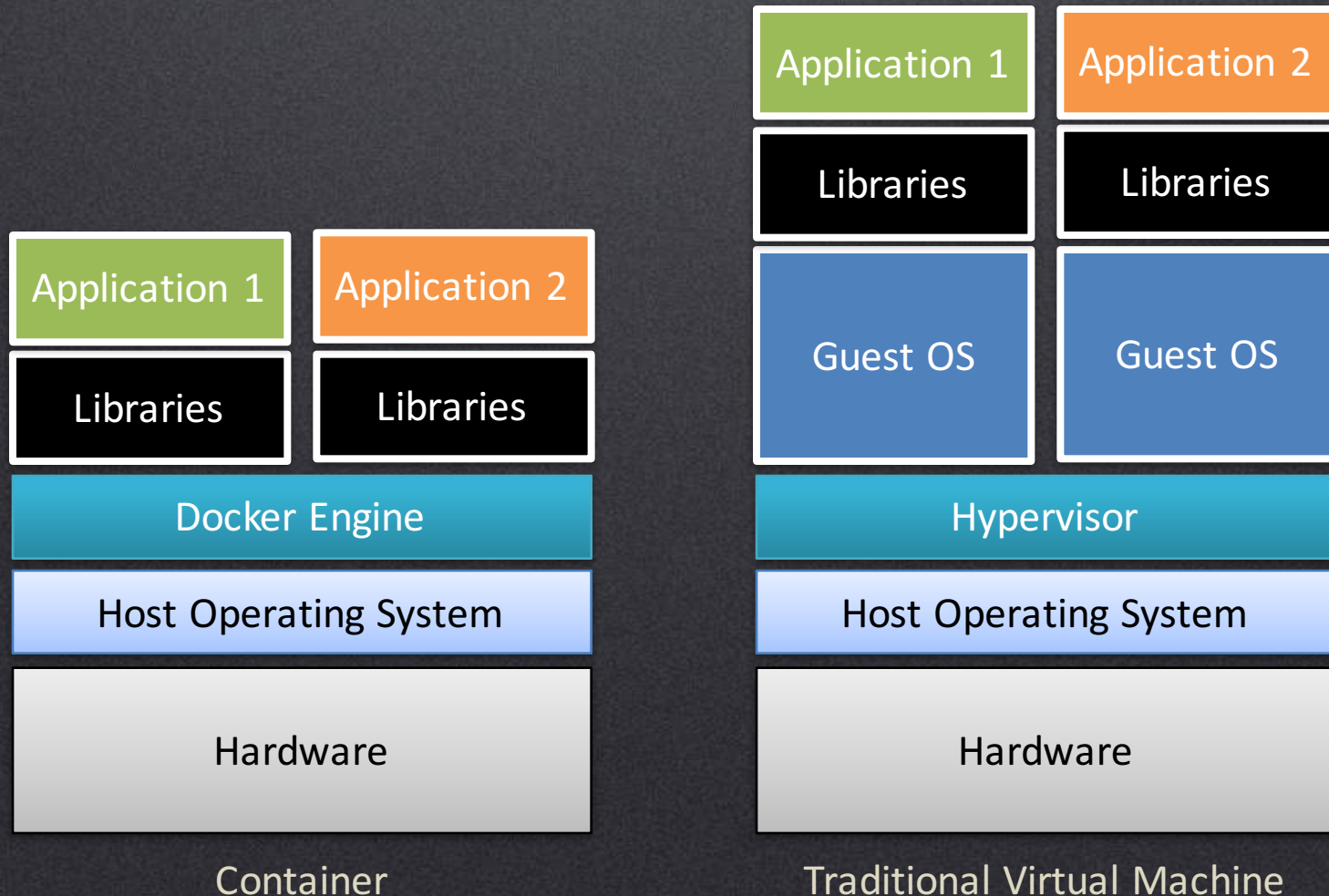
Containers vs Virtual Machines



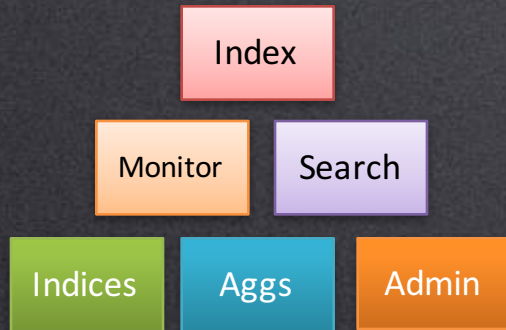
Containers vs Virtual Machines



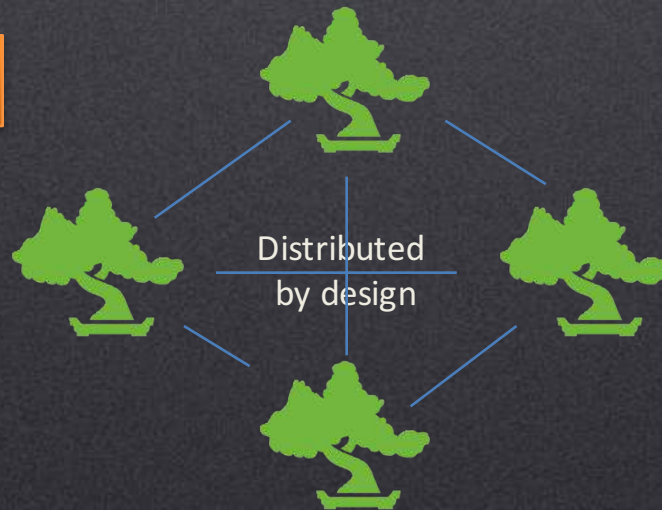
Containers vs Virtual Machines



Why Elasticsearch?



{ JSON }



Lucene



Running Official Elasticsearch Container

```
$ docker run -d elasticsearch
```


Running Official Elasticsearch Container

```
$ docker run -d elasticsearch:latest
```

Running Official Elasticsearch Container

```
$ docker run -d elasticsearch:latest
```

```
$ docker run -d --name es1 elasticsearch
```


Running Official Elasticsearch Container

```
$ docker run -d elasticsearch:latest
```

```
$ docker run -d --name es1 elasticsearch
```

```
$ docker run -d --name es1 -e ES_HEAP_SIZE=1g elasticsearch
```

Running Official Elasticsearch Container

```
$ docker run -d elasticsearch:latest
```

```
$ docker run -d --name es1 elasticsearch
```

```
$ docker run -d --name es1 -e ES_HEAP_SIZE=1g elasticsearch
```

```
$ docker run -d --name es1 elasticsearch -Dnode.name bbuzz
```


Container Constraints

```
$ docker run -d -m 2G elasticsearch
```

```
$ docker run -d -m 2G --memory-swappiness=0 elasticsearch
```

Container Constraints

```
$ docker run -d -m 2G elasticsearch
```

```
$ docker run -d -m 2G --memory-swappiness=0 elasticsearch
```

```
$ docker run -d --cpuset-cpus="1,3" elasticsearch
```


Container Constraints

```
$ docker run -d -m 2G elasticsearch
```

```
$ docker run -d -m 2G --memory-swappiness=0 elasticsearch
```

```
$ docker run -d --cpuset-cpus="1,3" elasticsearch
```

```
$ docker run -d --cpu-period=50000 --cpu-quota=25000 elasticsearch
```

Constraints - Good Practices

Limit container memory

Constraints - Good Practices

Limit container memory

Account for I/O cache when giving memory

Constraints - Good Practices

Limit container memory

Account for I/O cache when giving memory

Limit amount of CPU cores

Constraints - Good Practices

Limit container memory

Account for I/O cache when giving memory

Limit amount of CPU cores

Remember about JVM GC

Creating Optimized Image

Dockerfile:

```
FROM elasticsearch
```

```
ADD ./elasticsearch.yml /usr/share/elasticsearch/config/
```


Creating Optimized Image

Dockerfile:

```
FROM elasticsearch
```

```
ADD ./elasticsearch.yml /usr/share/elasticsearch/config/
```

```
$ docker build -t bbuzz/example .
```

Creating Optimized Image

Dockerfile:

```
FROM elasticsearch
```

```
ADD ./elasticsearch.yml /usr/share/elasticsearch/config/
```

```
$ docker build -t bbuzz/example .
```

Sending build context to Docker daemon 5.12 kB

Step 1 : FROM elasticsearch

---> 1e23f30a3667

Step 2 : ADD ./elasticsearch.yml /usr/share/elasticsearch/config/

---> 015f12adfd2a

Removing intermediate container de560c6ae0d1

Successfully built 015f12adfd2a

Dealing With Network

```
$ docker run -d -p 9200:9200 -p 9300:9300 elasticsearch
```

Dealing With Network

```
$ docker run -d -p 9200:9200 -p 9300:9300 elasticsearch
```

```
$ docker run -d --link es1 elasticsearch
```

```
-Ddiscovery.zen.ping.unicast.hosts=es1
```


Dealing With Network

```
$ docker run -d -p 9200:9200 -p 9300:9300 elasticsearch
```

```
$ docker run -d --link es1 elasticsearch
```

```
-Ddiscovery.zen.ping.unicast.hosts=es1
```

Add `network.publish_host` when building own container

Dealing With Network

```
$ docker run -d -p 9200:9200 -p 9300:9300 elasticsearch
```

```
$ docker run -d --link es1 elasticsearch
```

```
-Ddiscovery.zen.ping.unicast.hosts=es1
```

Add `network.publish_host` when building own container

Add `discovery.zen.ping.unicast.hosts` when building own
container

Network - Good Practices

Separate network for Elasticsearch cluster

Network - Good Practices

Separate network for Elasticsearch cluster

Common host names for containers

```
$ docker run -d -h esnode1 elasticsearch
```


Network - Good Practices

Separate network for Elasticsearch cluster

Common host names for containers

```
$ docker run -d -h esnode1 elasticsearch
```

Expose **only needed** ports

Network - Good Practices

Separate network for Elasticsearch cluster

Common host names for containers

```
$ docker run -d -h esnode1 elasticsearch
```

Expose **only needed** ports

Elasticsearch **data** & **client** nodes point to **masters** only

Dealing With Storage

By default in `/usr/share/elasticsearch/data`

Dealing With Storage

By default in `/usr/share/elasticsearch/data`

By default **not persisted**

Dealing With Storage

By default in `/usr/share/elasticsearch/data`

By default **not persisted**

```
$ docker run -d
```

```
  -v /opt/elasticsearch/data:/usr/share/elasticsearch/data  
  elasticsearch
```

Dealing With Storage

By default in `/usr/share/elasticsearch/data`

By default **not persisted**

```
$ docker run -d
```

```
  -v /opt/elasticsearch/data:/usr/share/elasticsearch/data  
  elasticsearch
```

Use **data only** Docker volumes



Permissions

Data Only Docker Volumes

Bypasses **Union File System**

Data Only Docker Volumes

Bypasses **Union File System**

Can be shared between containers

Data Only Docker Volumes

Bypasses **Union File System**

Can be shared between containers

Data volumes **persist** if the **container** itself is **deleted**

Data Only Docker Volumes

Bypasses **Union File System**

Can be shared between containers

Data volumes **persist** if the **container** itself is **deleted**

```
$ docker create -v /mnt/es/data:/usr/share/elasticsearch/data  
--name esdata elasticsearch
```



Permissions

Data Only Docker Volumes

Bypasses **Union File System**

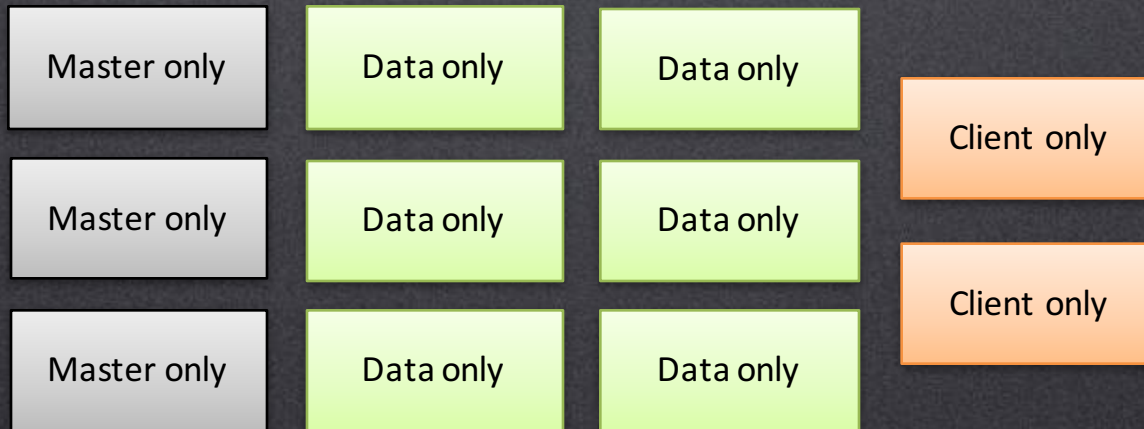
Can be shared between containers

Data volumes **persist** if the **container** itself is **deleted**

```
$ docker create -v /mnt/es/data:/usr/share/elasticsearch/data  
  --name esdata elasticsearch
```

```
$ docker run --volumes-from esdata elasticsearch
```

Highly Available Cluster



Highly Available Cluster



$\text{minimum_master_nodes} = N/2 + 1$

Master Nodes & Docker

```
$ docker run -d elasticsearch  
-Dnode.master=true  
-Dnode.data=false  
-Dnode.client=false
```

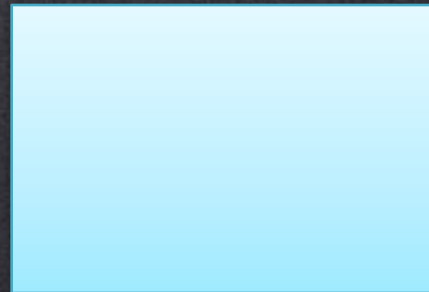
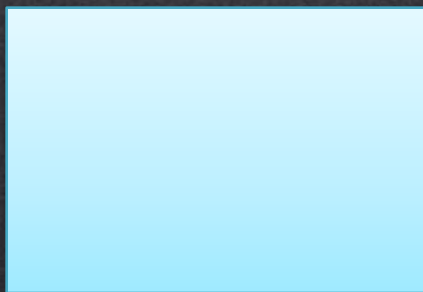
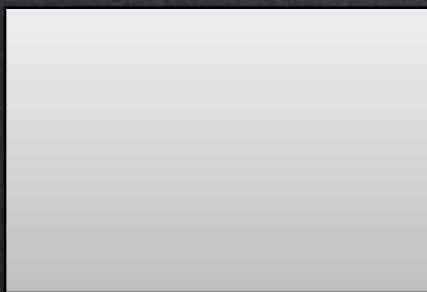
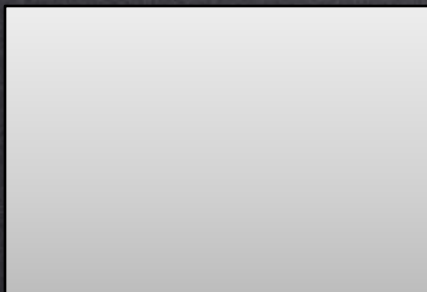

Client Nodes & Docker

```
$ docker run -d elasticsearch  
-Dnode.master=false  
-Dnode.data=false  
-Dnode.client=true
```

Data Nodes & Docker

```
$ docker run -d elasticsearch  
-Dnode.master=false  
-Dnode.data=true  
-Dnode.client=false
```


Multiple Tiers



node.tag=**hot**

node.tag=**cold**

node.tag=**cold**

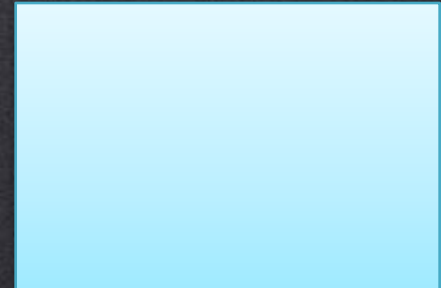
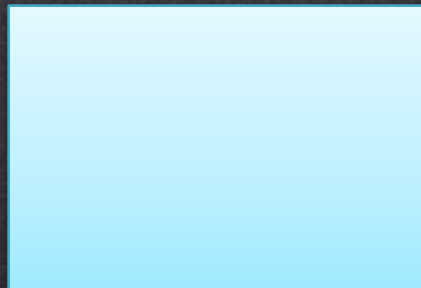
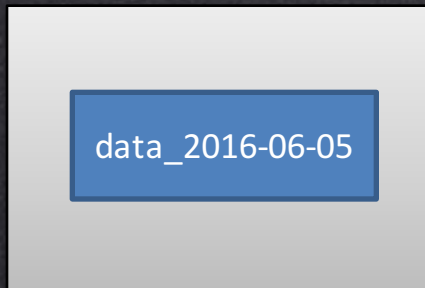
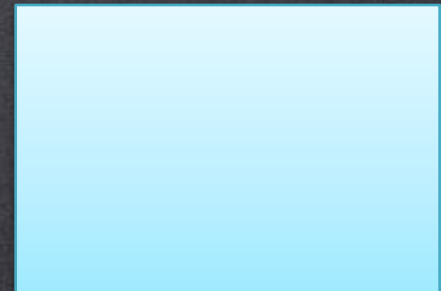
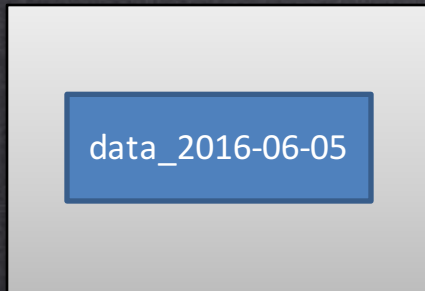
Multiple Tiers

```
$ docker run -d elasticsearch -Dnode.tag=hot
```


Multiple Tiers

```
curl -XPUT 'localhost:9200/data_2016-06-05' -d '{
  "settings": {
    "index.routing.allocation.include.tag" : "hot"
  }
}'
```

Multiple Tiers

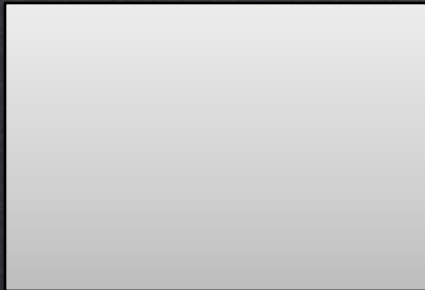


node.tag=**hot**

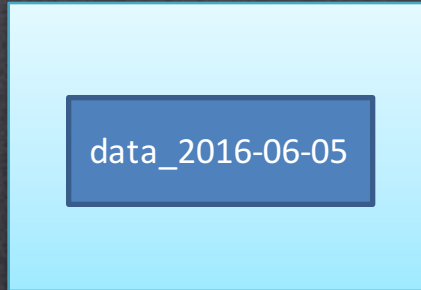
node.tag=**cold**

node.tag=**cold**

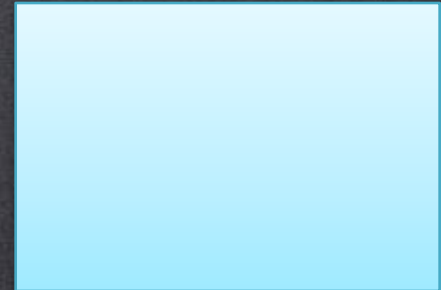
Multiple Tiers



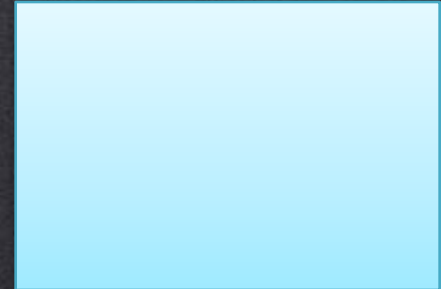
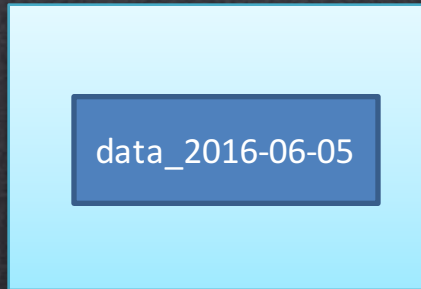
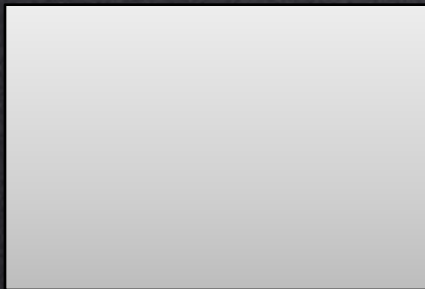
node.tag=**hot**



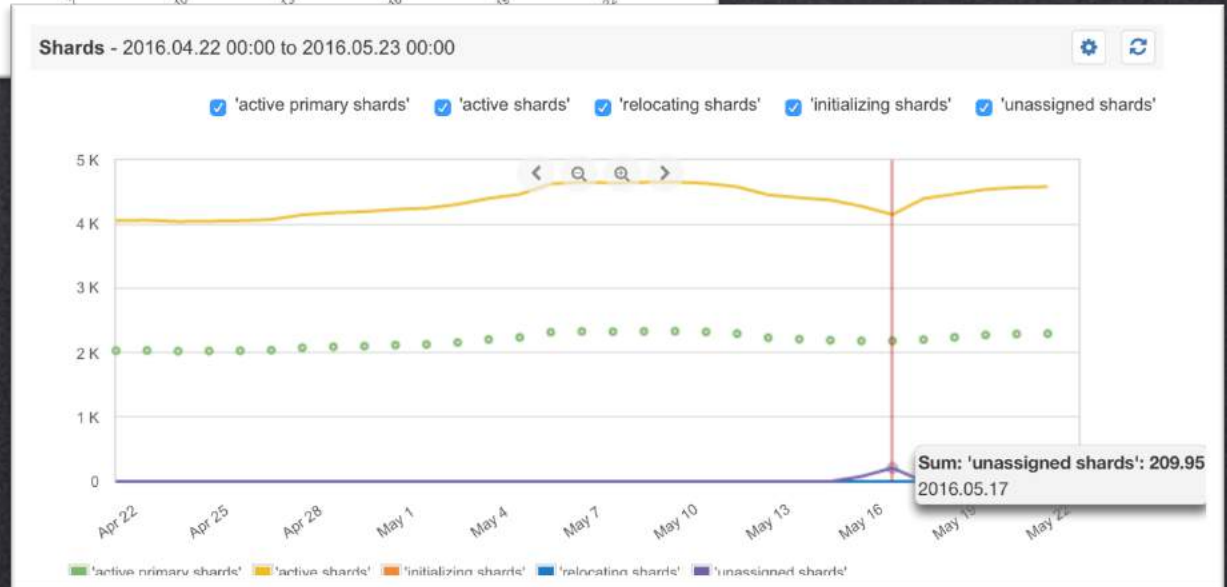
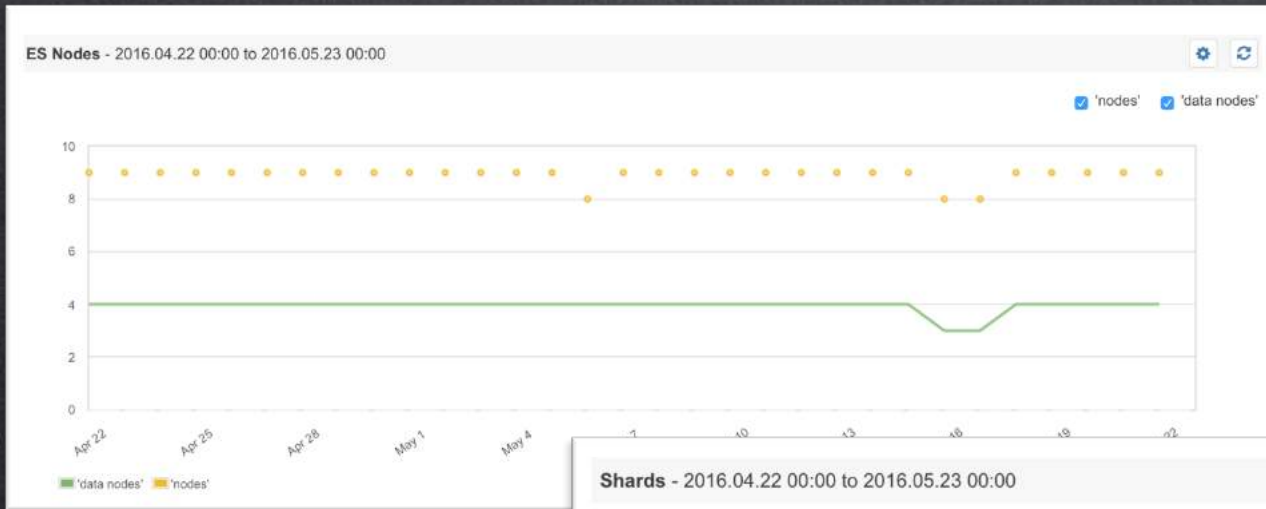
node.tag=**cold**



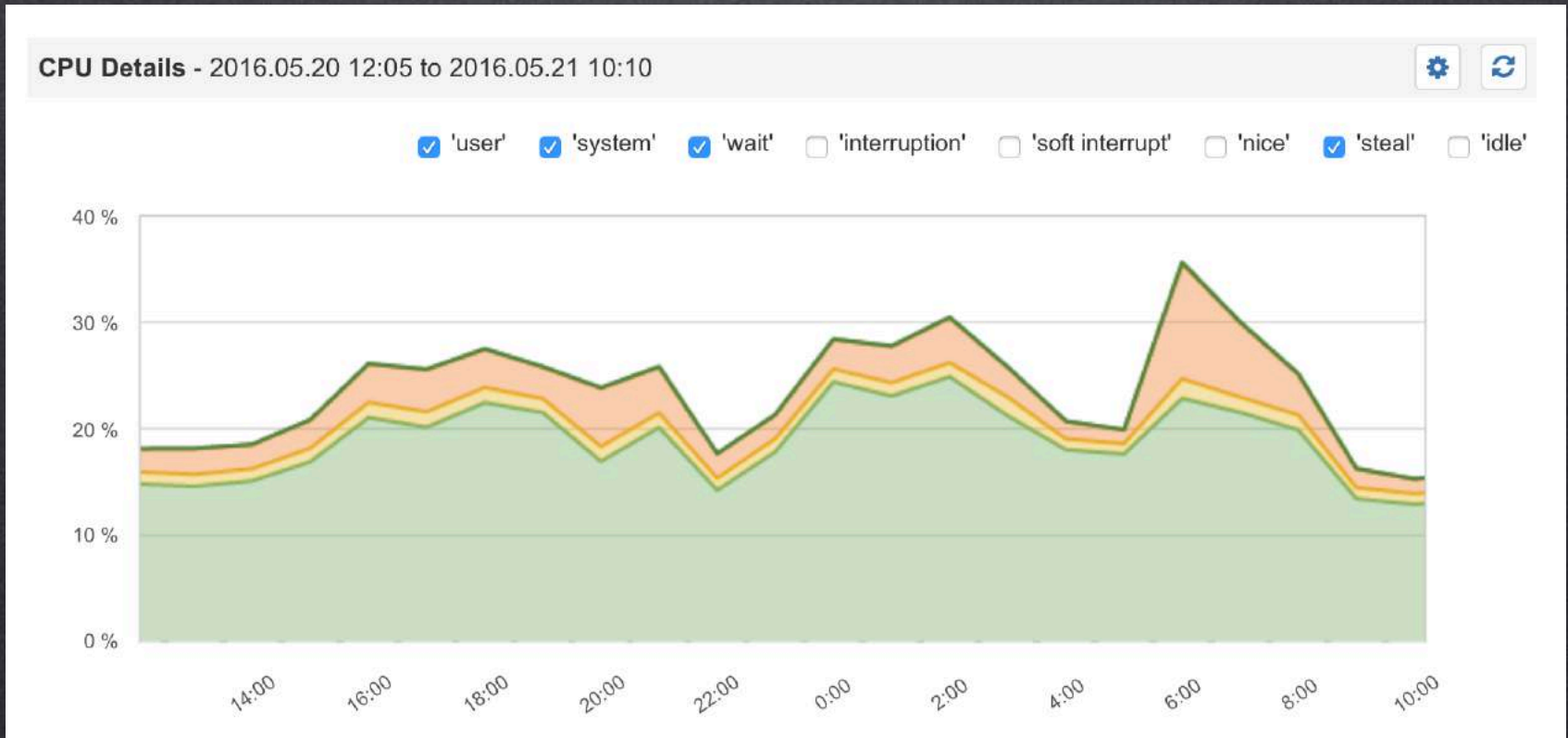
node.tag=**cold**



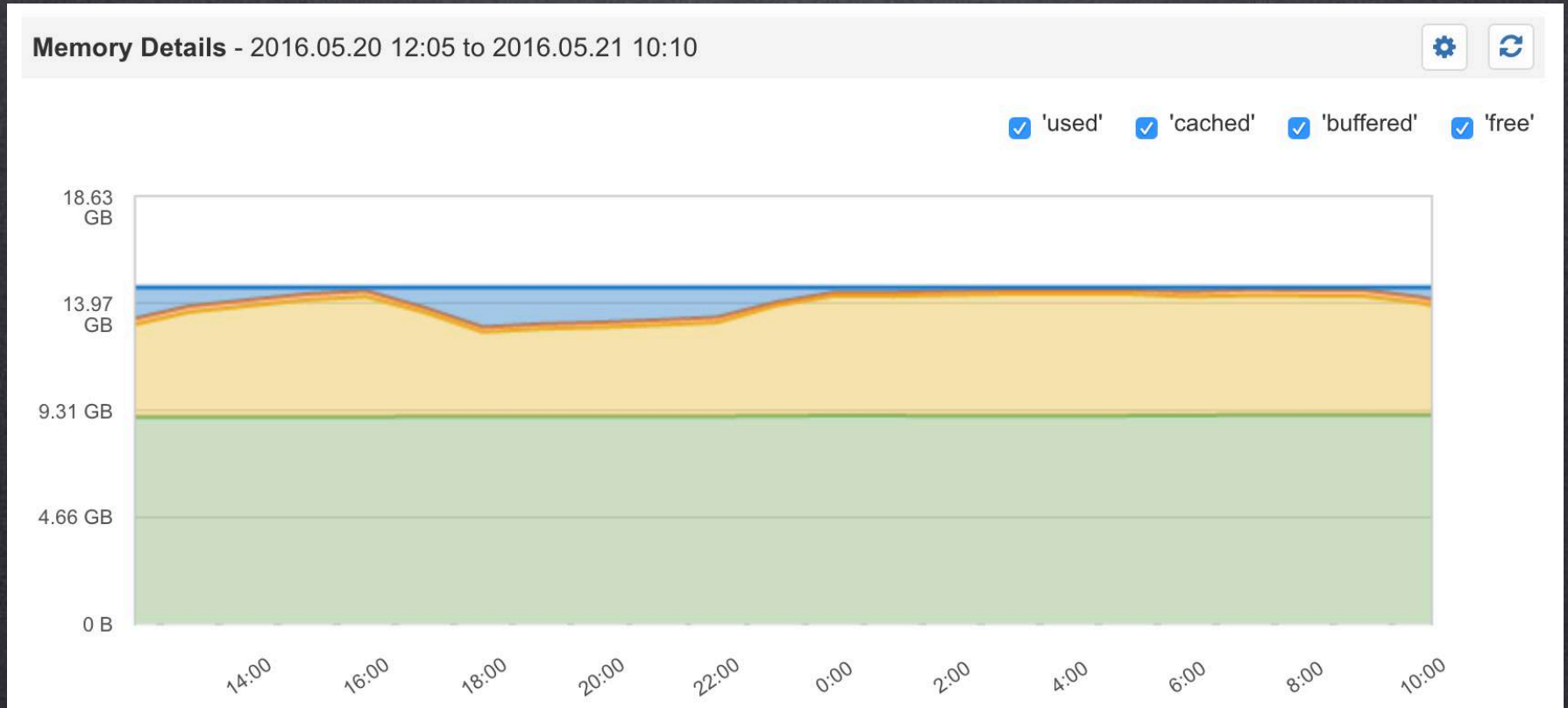
Top Metrics – Health & Shards



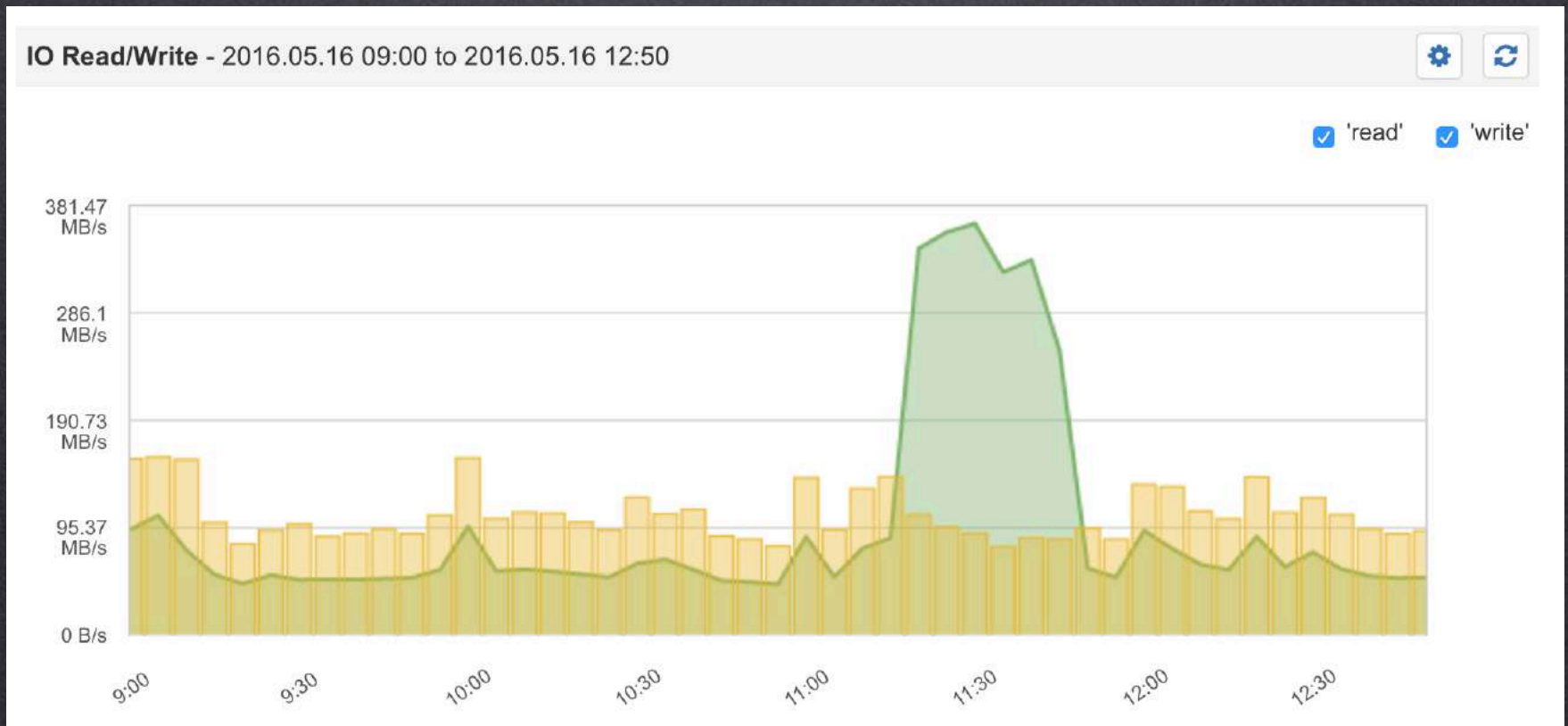
Top Metrics - CPU



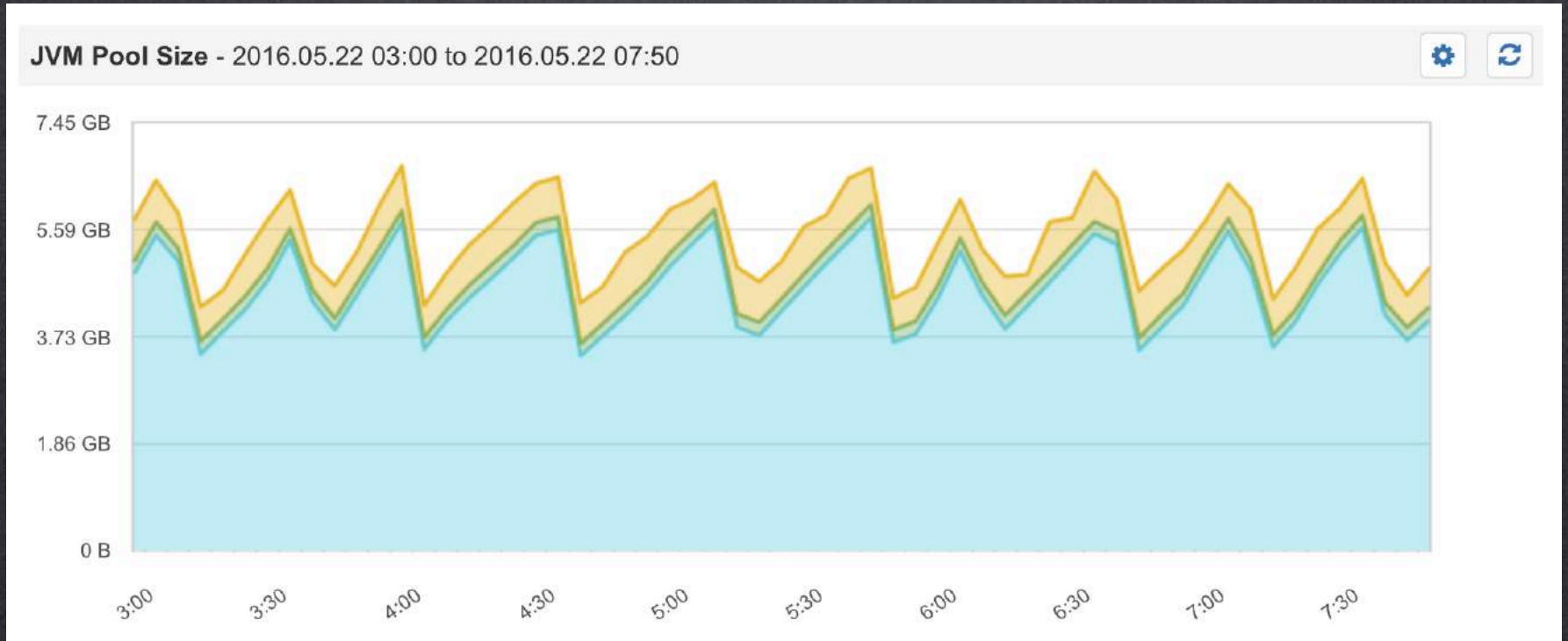
Top Metrics – Memory Usage



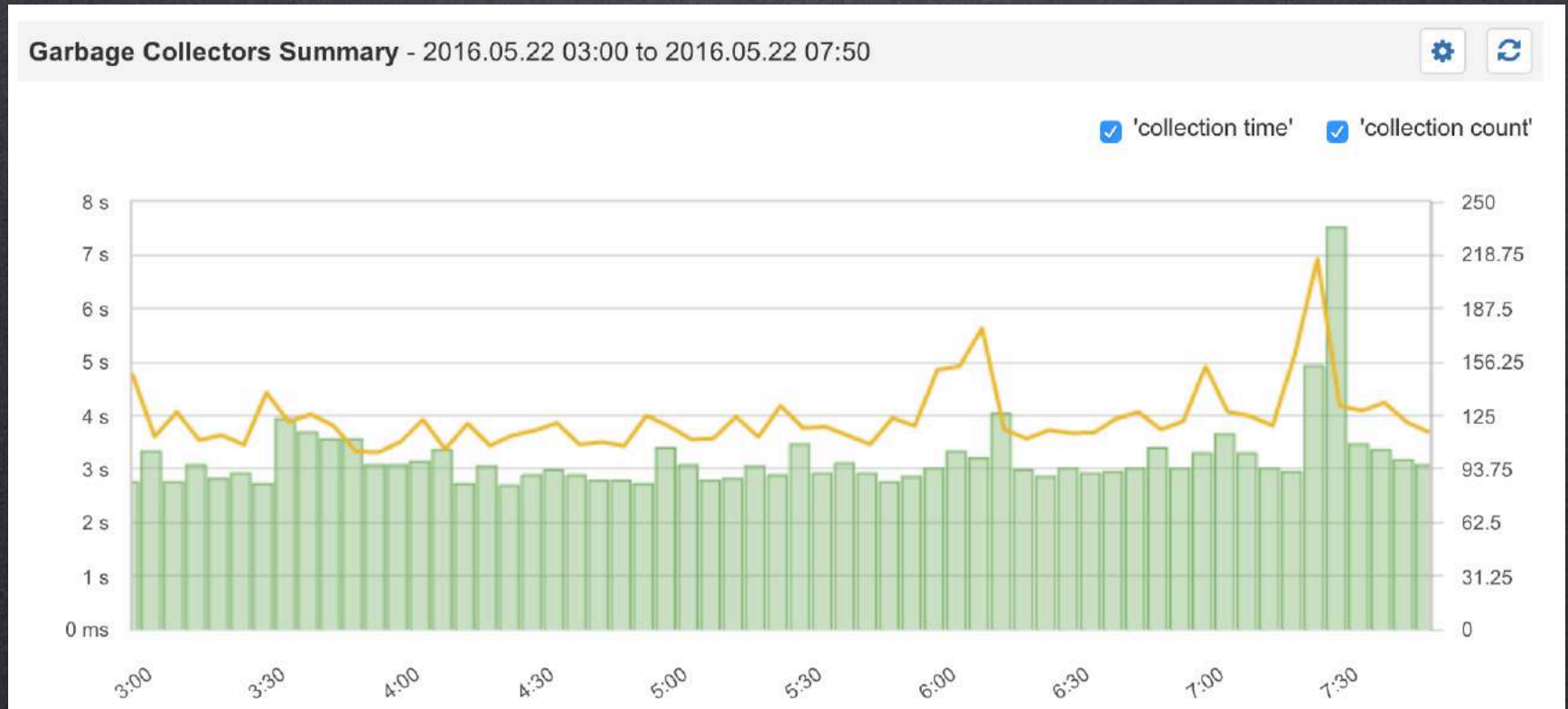
Top Metrics – I/O Usage



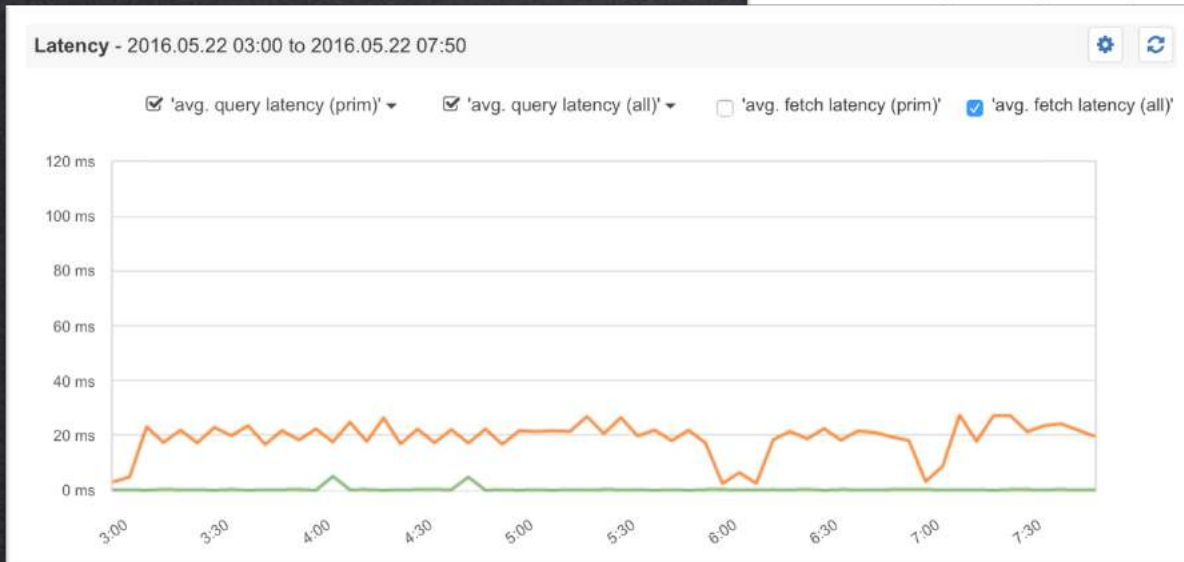
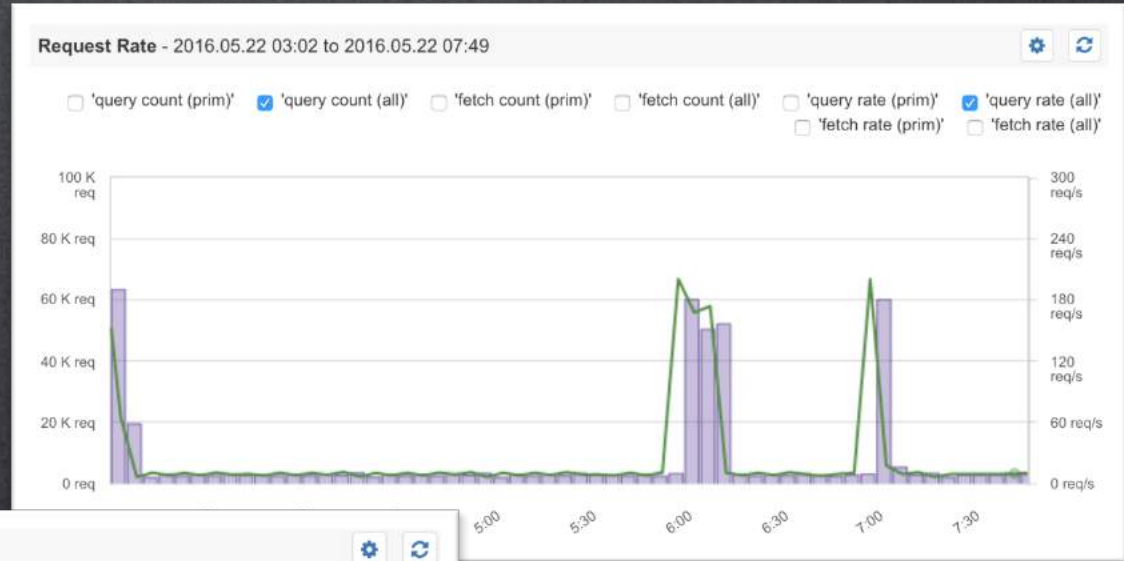
Top Metrics – JVM Heap



Top Metrics – Garbage Collector



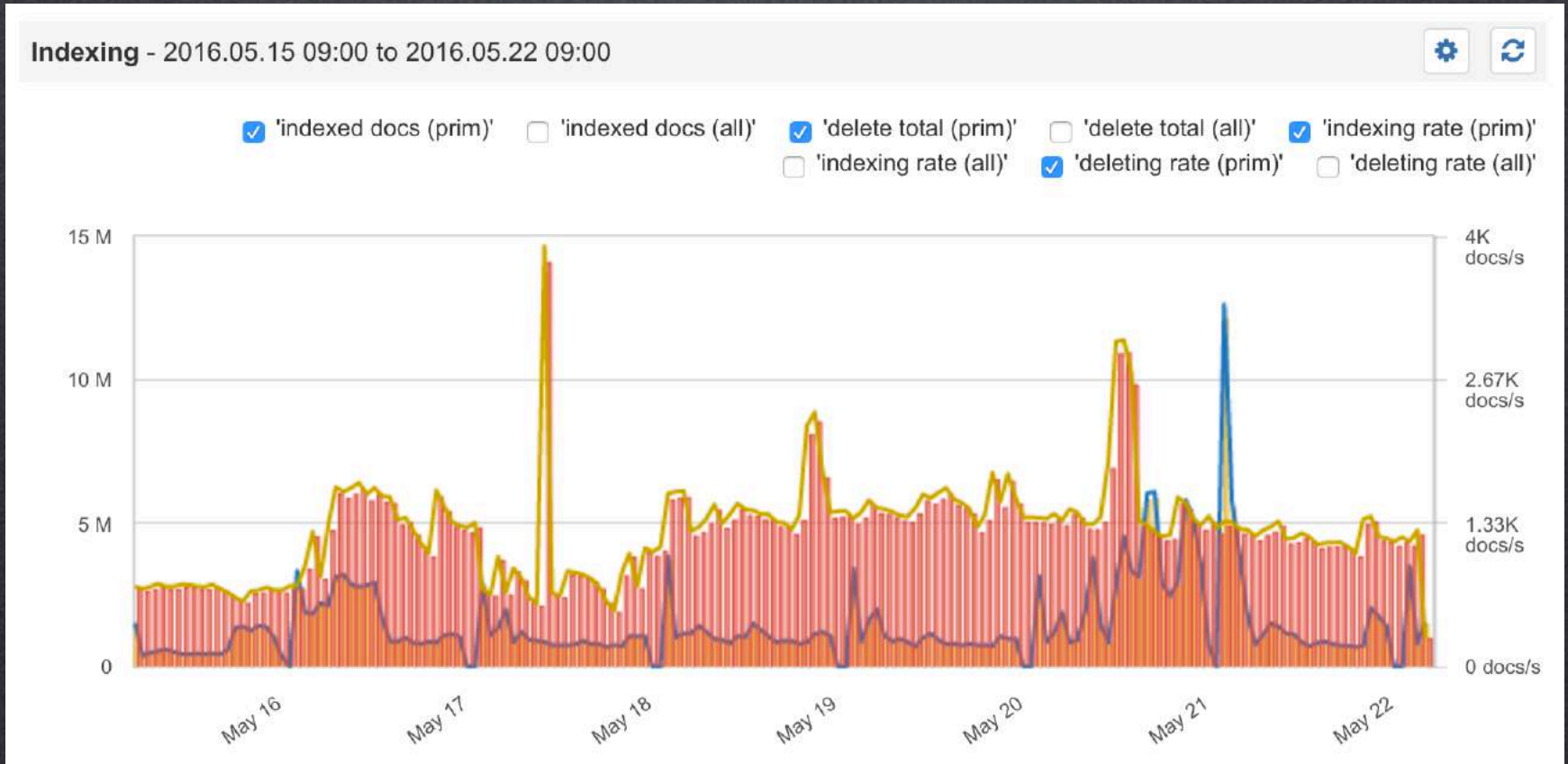
Top Metrics – Request Rate & Latency



Top Metrics - Caches



Top Metrics – Indexing

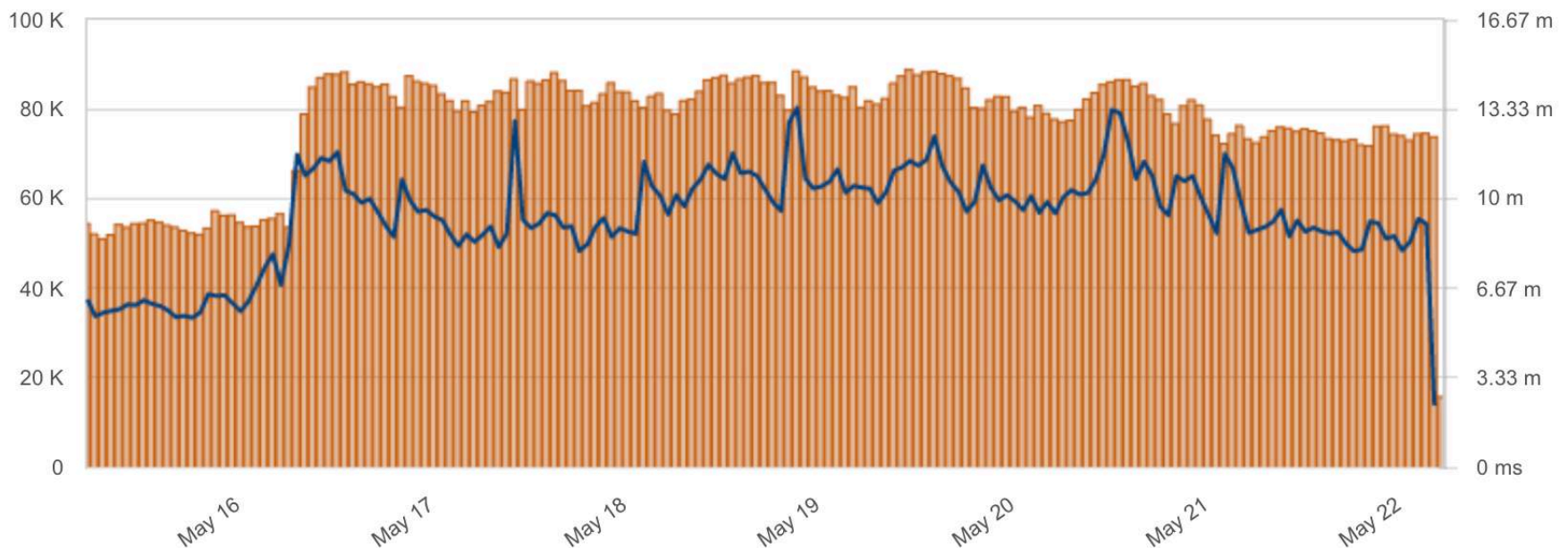


Top Metrics – Refresh Time

Refresh/Flush/Merge - 2016.05.15 09:00 to 2016.05.22 09:00



- 'refresh count (prim)' 'refresh count (all)' 'flush count (prim)' 'flush count (all)' 'merge count (prim)'
 'merge count (all)' 'refresh time (prim)' 'refresh time (all)' 'flush time (prim)' 'flush time (all)' 'merge time (prim)'
 'merge time (all)'

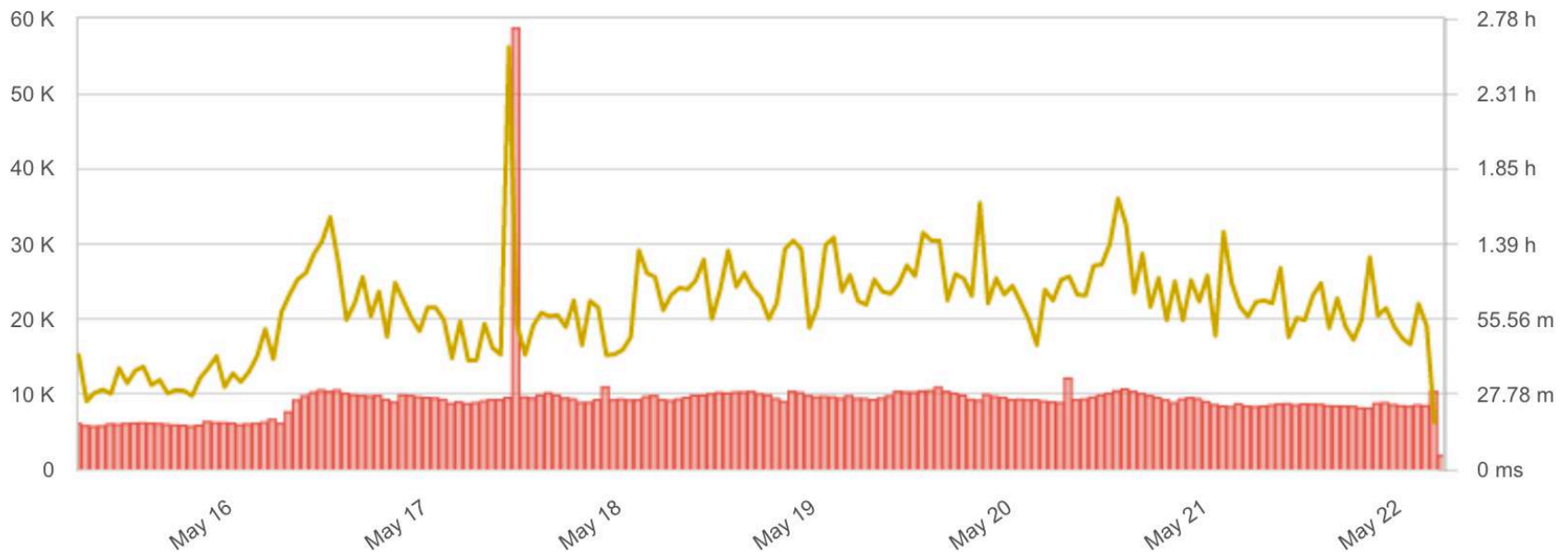


Top 10 Metrics – Merge Time

Refresh/Flush/Merge - 2016.05.15 09:00 to 2016.05.22 09:00



- 'refresh count (prim)'
- 'refresh count (all)'
- 'flush count (prim)'
- 'flush count (all)'
- 'merge count (prim)'
- 'merge count (all)'
- 'refresh time (prim)'
- 'refresh time (all)'
- 'flush time (prim)'
- 'flush time (all)'
- 'merge time (prim)'
- 'merge time (all)'



Short summary



We Are Hiring !

Dig Search ?

Dig Analytics ?

Dig Big Data ?

Dig Performance ?

Dig Logging ?

Dig working with and in open – source ?

We're hiring world – wide !

<http://sematext.com/about/jobs.html>

Thank You !

Rafał Kuć

@kucrafal

rafal.kuc@sematext.com

Sematext

@sematext

<http://sematext.com>

<http://blog.sematext.com>

